

**NATALI**

**Neural network Aerosol Typing Algorithm based on LIdar data**

**TD13 User guide**

**Datasheet on software performances, limitations and constraints**

**Issue 1, revision 03**

**November 8th, 2016**

|  |  |
| --- | --- |
| Project no.: | 4000110671/14/I-LG |
|  |  |
| Project coordinator: | Doina Nicolae |
|  | National Institute of R&D for Optoelectronics INOE |
|  | nnicol@inoe.ro |
|  |  |
| Technical officer: | Jonas von Bismarck |
|  | ESA/ESRIN |
|  | Jonas.Von.Bismarck@esa.int |

# Document Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| **Document, version** | **Date** | **Changes** | **Originator** |
| DOC, v00 | 31/03/2016 | Original version | Camelia Talianu, INOE |
| DOC, v01 | 26/04/2016 | Description of the voting procedure added | Camelia Talianu, INOE |
| DOC, v02 | 19/09/2016 | Updated for software’s new graphical interface. | Victor Nicolae, INOE |
| DOC, v03 | 8/11/2016 | Updated for the latest software version (1.1.5). | Victor Nicolae, INOE |

Table of contents

[Document Change Record 2](#__RefHeading___Toc449374988)

[1 Introduction 4](#__RefHeading___Toc449374989)

[1.1 Purpose 4](#__RefHeading___Toc449374990)

[1.2 Definitions, acronyms and abbreviations 4](#__RefHeading___Toc449374991)

[1.3 Applicable Documents 4](#__RefHeading___Toc449374992)

[1.4 Reference Documents 4](#__RefHeading___Toc449374993)

[1.5 Structure of the document 5](#__RefHeading___Toc449374994)

[2 Application 5](#__RefHeading___Toc449374995)

[3 Algorithm description 6](#__RefHeading___Toc449374996)

[3.1 Modules 8](#__RefHeading___Toc449374997)

[3.1.1 Input module 8](#__RefHeading___Toc449374998)

[3.1.2 Typing module 10](#__RefHeading___Toc449374999)

[3.1.3 Output module 12](#__RefHeading___Toc449375000)

[4 Software tool description 12](#__RefHeading___Toc449375001)

[4.1 Structure 12](#__RefHeading___Toc449375002)

[4.1.1 natali.py 12](#__RefHeading___Toc449375003)

[4.1.2 nt\_input.py 13](#__RefHeading___Toc449375004)

[4.1.3 nt\_typing.py 13](#__RefHeading___Toc449375005)

[4.1.4 nt\_output.py 14](#__RefHeading___Toc449375006)

[4.2 Software requirements 14](#__RefHeading___Toc449375007)

[4.3 Installation 14](#__RefHeading___Toc449375008)

[4.3.1 Python 2.7 15](#__RefHeading___Toc449375009)

[4.3.2 NetCDF4 C library 16](#__RefHeading___Toc449375010)

[4.3.3 numpy Python module 16](#__RefHeading___Toc449375011)

[4.3.4 netcdf4-python module 16](#__RefHeading___Toc449375012)

[4.4 Usage 16](#__RefHeading___Toc449375013)

[4.5 Output files 18](#__RefHeading___Toc449375014)

[4.5.1 The log file 18](#__RefHeading___Toc449375015)

[4.5.2 The .TXT file 19](#__RefHeading___Toc449375016)

[4.5.3 The .CSV file 22](#__RefHeading___Toc449375017)

[5 Datasheet on software performances, limitations and constraints 24](#__RefHeading___Toc449375018)

[6 Copyright and disclaimer notice 28](#__RefHeading___Toc449375019)

[6.1 Copyright notice 28](#__RefHeading___Toc449375020)

[6.2 Disclaimer notice 28](#__RefHeading___Toc449375021)

[6.3 Software usage policy 28](#__RefHeading___Toc449375022)

# Introduction

## Purpose

This document describes the algorithms and the software tool for retrieving the type of aerosol from multiwavelength optical data.

## Definitions, acronyms and abbreviations

ANN – Artificial Neural Network

JN - Jordan /Elman network

GFF – Generalized Feed Forward network

LIDAR – Light detection and ranging

EARLINET - European Aerosol Research LIdar NETwork

## Applicable Documents

Contract number: 4000110671/14/I-LG

Project proposal: Neural network Aerosol Typing Algorithm based on LIdar data (NATALI), Reference: AO7557-NATALI

## Reference Documents

NeuroSolutions software manuals

TD01 Algorithm Theoretical Basis Document

TD03 List of input parameters considered, and ranges

TD04 Table list of the software technical requirements

TD06 Schematics of the neural network

TD07 Report on software design

TD09 Report on software performance after the learning process

TD10 Report on sensitivity and blind tests

Belegante, L., Nicolae, D., Nemuc, A., Talianu, C. and Derognat, C., Retrieval of the boundary layer height from active and passive remote sensors. Comparison with a NWP model, *Acta Geophysica*, 62(2), 276–289, doi:10.2478/s11600-013-0167-4, 2014.

Dubovik, O., Sinyak, A., Lapyonok, T., Holben, B. N., Mishchenko, M., Yang, P., Eck, T. F., Volten, H., Mu˜noz, O., Veihelmann, B., van der Zande, W. J., Leon, J.-F., Sorokin, M., and Slutsker, I., Application of spheroid models to account for aerosol particle nonsphericity in remote sensing of desert dust, *J. Geophys. Res.*, 111, D11208, doi:10.1029/2005JD006619, 2006.

Hess, M., P. Koepke, and I. Schult, Optical properties of aerosols and clouds: The software package OPAC, *Bull. Am. Meteorol. Soc.*, 79, 831 – 844, 1998.

Koepke, P., M. Hess, I. Schult, and E. P. Shettle, Global aerosol data set, *MPI Meteorologie Hamburg* Report No. 243, 44 pp, 1997

Munoz, O., Volten, H., de Haan, J. F., Vassen, W., and Hovenier, J. W., Experimental determination of scattering matrices of randomly oriented fly ash and clay particles at 442 and 633 nm, *J. Geophys. Res.*, 106, 22833–22844, 2001.

## Structure of the document

Section 2 of the document describes the application for which the software was designed and implemented. Algorithms are presented in Section 3, and the software tool in Section 4, including structure, requirements, installation and usage. A summary of software performances, limitations and constraints is provided in Section 5.

# Application

Inputs parameters are typical data products from EARLINET (European Aerosol Reserach LIdar NETwork) database:

* () backscatter (180o) coefficient profiles at 1064, 532 and 355 nm
* () extinction coefficient profiles at 532 and 355 nm
* () (optional) linear particle depolarization profile at 532 nm

Output consists in the most probable aerosol type within the layers. Depending on the physical content (with or without depolarization), and of the quality of the optical data (calibration, uncertainty), typing is performed considering:

* high resolution typing (AH): 13 aerosol types (pure, mixtures of 2, and mixtures of 3 pure aerosol types) if all optical parameters are provided with good quality
* low resolution typing (AL): 6 predominant aerosol types (pure with max. 30% traces of other types) if all optical parameters are provided but the uncertainty is high
* low resolution typing (BL): 5 predominant aerosol types (pure with max. 30% traces of other types) if particle depolarization is missing

The aerosol types retrieved are summarized in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Aerosol mixture** | **High resolution typing****(AH)** | **Low resolution typing with depolarization****(AL)** | **Low resolution typing without depolarization****(BL)** |
| Continental | Continental | Continental | Continental |
| Dust | Dust | Dust | Dust |
| Continental polluted | Continental polluted | Continental polluted | Continental polluted |
| Marine | Marine | Marine | Marine |
| Smoke | Smoke | Smoke | Smoke |
| Continental + Dust | Continental dust | Continental / Dust | Continental / Dust |
| VolcanicDust + MarineVolcanic + Marine | Mineral mixtures/Volcanic | Dust / Marine | Dust / Marine |
| Continental + Smoke | Continental smoke | Continental polluted / Smoke | Continental polluted / Smoke |
| Dust + Smoke | Dust polluted | Dust / Smoke | Dust / Smoke |
| Continental + Marine | Coastal | Continental / Marine | Continental / Marine |
| Continental polluted + Marine | Coastal polluted | Continental polluted / Marine | Continental polluted / Marine |
| Continental + Dust + Marine | Mixed dust | Dust / Marine | Dust / Marine |
| Continental + Smoke + Marine | Mixed smoke | Continental polluted / Smoke | Continental polluted / Smoke |

Table 1 Aerosol types retrieved for various situations: in black – types retrieved directly; in cyan – corresponding predominant types

**Note**: The names of the aerosol types are conventions which account for:

|  |  |  |
| --- | --- | --- |
| **Name of the aerosol type** | **Source** | **Characteristics of the particles** |
| Continental | land surfaces | medium-size, medium-spherical, medium absorbing |
| Dust | desert surfaces | big, non-spherical, medium absorbing |
| Continental polluted | industrial sites | small, spherical, highly absorbing |
| Marine | sea surface | big, spherical, low absorbing |
| Smoke | vegetation fires | small, spherical, highly absorbing |
| Volcanic | volcanoes | big, non-spherical, highly absorbing |
| Mixtures | mixed | combinations of the above |

Table 1 Conventions for the names of the aerosol types

**Warning!** Residuals from clouds in the aerosol input data (backscatter, extinction and depolarization profiles) may lead to an incorrect classification: Marine or mixtures with Marine. As such, the potential “cloud corruption” has been added to all types containing marine particles, e.g. Marine/CC. In case such types are present in the output data, the user should check the possibility to have residuals from clouds in the input files.

**Warning!** Volcanic cannot be distinguished from mineral mixtures based on the optical properties from lidar, especially when linear article depolarization is not available. As such, the Volcanic type is always provided as Mineral mixtures. Additional information will be necessary in order to identify volcanic ash.

# Algorithm description

The algorithm relies on a set of Artificial Neural Networks which are trained to recognize the aerosol type based on a set of input optical data. The optical data has to be characteristic for a certain type of particles (i.e. to be independent on the density of the particles) , therefore the 3 +2 (+1) lidar data are first used to compute the intensive properties such as Angstrom exponent, color ratios, color indexes and lidar ratios. These are further used by the ANNs for classification. The ability of the ANNs to retrieve aerosol type depends strongly on the physical content of the optical inputs, as well as on their uncertainty.

Each ANN was trained on a comprehensive set of 50000 synthetic cases obtained from a specially designed aerosol model. Aerosols are simulated as an externally mixture of basic components (water soluble, insoluble, soot, mineral - nucleation, accumulation, coarse, sulfates, sea salt - accumulation, coarse) in different proportions, and at various values of the humidity (40% … 90%). Microphysical properties of the components were picked up from GADS (Global Aerosol Database, Koepke et al., properties of the components were picked up from GADS (Global Aerosol Database, Koepke et al., 1997), variation with the relative humidity from OPAC (Optical properties of Aerosol and Clouds, Hess et al 1998), while the overall optical properties were computed using T-matrix calculations and typical asphericities from literature (Munoz et al., 2001; Dubovik et al., 2006). Pure types were mixed following a linear progression of the number densities. Out of all combinations possible, the most representative mixtures were selected (see Table 1). A 20% relative error was considered for the intensive optical parameters of the aerosol types, which is also the limit expected for lidar data (beyond this value, the typing is uncertain).

To increase the confidence of the typing, 3 ANNs with different structures were developed for each case (AH, AL, BL):

* Jordan /Elman Network with 6 hidden layers and Momentum learning rule
* Jordan /Elman Network with 8 hidden layers and Conjugate gradient learning rule
* Generalized feedforward network with 6 (A class) or 10 hidden layers (B class)

Supervised training has been used to train all ANNs, sets of input and output parameters being successively presented to the network for around 1000 epochs per training cycle. Backpropagation learning is used; the weights are changed based on their previous value and a correction term.

Jordan /Elman network (JE) represents an extension of the multilayer perceptron network with processing elements that remembers the past activity, the context units. Two different approaches are implemented: the Elman network, where the activity of the first processing elements are written to the context units, and the Jordan network, which copies the output of the network. These ANN are layered feedforward networks typically trained with static backpropagation and can approximate any input/output map. The Jordan/Elman networks used for aerosols typing have 6 hidden layers with 50 processing elements per layer for fist 4 layers; 45 processing elements- fifth layer; 37 processing elements – sixth layer, Tanh axons, 1000 iterations over the training set. Either Momentum learning or Conjugate gradient are used for JE networks.

The Momentum learning rule is used, which provides the gradient descent with some inertia, depending on the momentum parameter, which gives the smoothness of the gradient estimation. The Conjugate gradient learning rule has no parameters to be adjusted, like learning rates or momentum parameter, and is faster and more accurate in respect with standard backpropagation.

A good percentage of training per aerosol class and stable performances and approximatively constant for all aerosols classes are the main advantages of JE. The disadvantages of JE are: slow training, limited performance improvement after training and reaches the training limit rapidly.

Generalized feedforward networks (GFF) represent a generalization of the multilayer perceptron, each layer feeds forward to all subsequent layers. The connections between axons/layers can jump over one or more layers. The GFF networks used for aerosols typing have 6 hidden layers (A class) or 10 hidden layers (B class), Tanh axons, 1000 iterations over the training set. In this case only the Momentum learning rule has been used.

The GFF networks trains rapidly and have low error of training after 2 training cycles. The main disadvantages are: trains efficiently only several cycles, a further improvement of weights cannot be considered and stable active performances per aerosol type overall, but lower values for several classes.

A schematics of the algorithm is presented below.



Figure 1 Schematics of the algorithm for aerosol typing

## Modules

NATALI is build on three modules:

* Input module: to prepare the inputs in the specific format of the ANNs
* Typing module: to run the ANNs and decide on the most probable aerosol type
* Output module: to save the results and logs

### Input module

The input module reads the lidar files in EARLINET NetCDF format, checks for the availability of all required parameters (1064, 532, 355, 532, 355, and optionally 532), identifies the layer geometrical boundaries, calculates the intensive optical parameters within each layer, their mean value and associated uncertainty.

The steps performed by the input module are:

* read NetCDF data in EARLINET format
* interrogate what parameters are included
	+ if depolarization is included, select class “A” for the ANNs
	+ if depolarization is not included, select class “B” for the ANN
* if a parameter is missing (except depolarization), reject the data
* identify layer bottoms and layer tops
* calculate profiles of intensive optical parameters
* calculate mean layer intensive optical parameters and associated uncertainties
* generate N values between the error limits, for each layer and for each parameter
* generate combinations between the values of the intensive optical parameters, for each layer
* convert the datasets corresponding to each layer in the ANN specific format

Layer boundaries are calculated by applying the gradient method on the 1064 nm backscatter coefficient profile (Belegante et al., 2014). The inflexion points of the second derivative of the profile data (computed with the Savistky-Golay filter) give the tops and the bottoms of the layer. Gross or fine structure of the aerosol layers is revealed by a higher or lower value of the smoothing parameter (adjustable) *FINES*S*E*. Only layers with a thickness larger than 300 m are considered relevant, for the reason of significant signal-to-noise ratio.

The intensive optical parameters and their associated uncertainties are computed for the middle part of each layer for which the signal-to-noise ratio is highest (no less than 200 m mid-layer), to exclude the margins which are affected by the smoothing:

Angstrom coefficient:

$AE\_{{355}/{532}}=-\frac{ln\left({α\_{355}}/{α\_{532}}\right)}{ln\left({355}/{532}\right)}$

UV/VIS Color ratio:

$CR\_{{355}/{532}}{=β\_{355}}/{β\_{532}}$

VIS/IR Color ratio:

$CR\_{{532}/{1064}}{=β\_{532}}/{β\_{1064}}$

UV/VIS Color index:

$CI\_{{355}/{532}}=-\frac{ln\left({β\_{355}}/{β\_{532}}\right)}{ln\left({355}/{532}\right)}$

VIS/IR Color index:

$CI\_{{532}/{1064}}=-\frac{ln\left({β\_{532}}/{β\_{1064}}\right)}{ln\left({532}/{1064}\right)}$

UV Lidar ratio:

$LR\_{355}{=α\_{355}}/{β\_{355}}$

VIS Lidar ratio:

$LR\_{532}{=α\_{532}}/{β\_{532}}$

The linear particle depolarization ratio is picked-up directly from the .b532 file, if existing. For each layer, and for all the above arrays, the module calculates averages and associated uncertainties.

Several filters are applied on the data, and only layers which pass these criteria are further considered for typing:

* availability of all necessary intensive optical parameters
* values of the intensive optical parameters are between acceptable limits (see Table 1)

|  |  |  |
| --- | --- | --- |
| **Intensive parameter** | **Min. acceptable value** | **Max. acceptable value** |
| Angstrom coefficient | - 2  | 6 |
| Color ratio | -2  | 6 |
| Color index | -2  | 6 |
| Lidar ratio (sr) | 5  | 200 |
| Linear particle depolarization ratio (%) | 0  | 60 |

Table 1 Acceptable limits for the layer average intensive optical parameters

**Note:** The acceptable ranges are exaggerated compared to the literature (e.g. color ratio should generally be in the 0 … 2 interval). The extension of the acceptable intervals was implemented in order to accommodate also not perfectly calibrated data. The ANNs are still capable to return the aerosol type if one or two intensive optical parameters are not perfect. However, the chances to return “Unknown” increase.

**Note:** In case that the relative error of any of the intensive optical parameters is higher than 20%, typing is performed but the result is flagged with “*Typing uncertain: relative error of intensive parameters [...] higher than 20%*” message.

For each layer and for each intensive optical parameter, the module generates a number of values (N, adjustable) between [average – uncertainty] and [average + uncertainty]. Data are than scrambled considering that any combination has a similar probability to describe the reality.

The cluster of possible combination of intensive optical parameters is prepared for the ANN input format.

### Typing module

The typing module runs in parallel the ANNs for each dataset representing a layer, and applies the voting procedure to identify the most probable aerosol type.

In case depolarization is available, the module runs in parallel 6 ANNs: 3 for high resolution (A1H, A2H, A3H) and 3 for low resolution typing (A1L, A2L, A3L). The probable aerosol type is provided by the high resolution ANNs, while the predominant type is provided by the low resolution ANNs. As such, if typing in high resolution fails (for reasons of quality of the data), the user has still access to some information, in low resolution.

If the depolarization is not available, the module runs in parallel 3 ANNs (B1L, B2L, B3L), and returns the most probable predominant aerosol type. By comparison to the low typing when depolarization is available, in this case the ANNs cannot distinguish the Volcanic type, as it overlaps completely (in all existing parameters) with Dust or Continental polluted. As such, only 5 predominant types are retrieved.

The voting procedure is applied in order to advise the user on the selection of the most probable answer, out of the 3 outputs from the ANNs. In principle, the answer with the highest trust level is selected. The trust level is computed as the weighted sum of the **confident answers’ probabilities** (which is an indication of the confidence with which each ANN was able to return the answer) and the **confident answer count** (which is an indication of the stability of the answer over the error interval). The weights are the same - 50% - for each member. In particular situations (e.g. when 1 or 2 ANNs return “unknown”), the valid answer (different than “unknown”) is accepted regardless the trust level.

The steps performed by the typing module are:

* read input datasets prepared by the Input module
* for each layer:
	+ select the ANN class(es) suited for the data (AH and AL, or BL)
	+ run the ANNs (6 or 3 in parallel)
	+ filter the outputs
		- for each ANN, select only the answers with a confidence better than 70% (adjustable)
		- for each ANN, select only the answers which agree on the type for more than 25% cases presented
	+ vote between ANNs in the same class (AH, AL, BL)
		- if all 3 ANNs return “unknown”, type is "unknown"
		- if 2 ANNs return “unknown” and the third return a type, accept the type
		- if 1 ANN returns “unknown” and the other 2 return the same types, accept the type
		- if 1 ANN returns “unknown” and the other 2 return different types, accept the result for which the trust level is higher
		- if there are more networks with the same trust level choose the one with the highest count of confident answers (more stable)
		- if there are more networks with the same confident answer count, choose the one with the highest overall confidence (more confident)
		- if all ANNs return different types, accept the result for which the trust level is higher
		- if there are more networks with the same trust level choose the one with the highest count of confident answers (more stable)
			* if there are more networks with the same confident answer count, choose the one with the highest overall confidence (more confident)

### Output module

The output module prepares and saves the files in 2 formats: CSV and human-readable (telegrams), and writes the log.

The .CSV file and the telegrams contain (different formats):

* identification of the datasets for which the typing was performed
* for each identified layer:
	+ geometrical top and bottom
	+ intensive optical parameters and associated uncertainties
	+ aerosol type retrieved by each ANN, the confidence and the number of agreements
	+ the most probable type selected with the voting procedure (in low and high resolution separately, if is the case)
	+ comments (generally referring to situations when optical data did not passed the quality criteria, or errors in the retrieval procedure)

Additional information (e.g. run time, run parameters, network error messages) is included in the telegrams.

# Software tool description

## Structure

The software’s code is structured in several modules which do all the work:

* A data processing module**: nt\_data.py**
* A typing module**: nt\_typing.py**
* An output module**: nt\_output.py**
* A graphical user interface module**: nt\_ui.py**

There are also two more helper Python modules in the source code: **natali.py** which is used to start the NATALI application and **nt\_globals.py** which contains the default values for parameters and various utilities.

### natali.py

This is the entry point to the program. It starts the graphical user interface, which will orchestrate the data and typing modules, as well as it starts the output printer which will be used later by the rest of the modules.

### nt\_data.py

The***LidarMeasurement***class inside the *nt\_data.py* file exposes methods to read data from the NetCDF files, identifies the aerosol layers, calculates the intensive parameters from the extensive ones and computes the average value and uncertainties inside each layer.

For each lidar dataset, the script constructs a *LidarMeasurement* object and calls its data processing methods in the following order:

measurement=LidarMeasurement(group\_name, measurements\_folder)

measurement.read\_data()

layers=measurement.get\_layers()

measurement.compute\_intensive\_parameters(layers=layers)

The *LidarMeasurement* class includes also methods to calculate the extensive and the intensive parameters within the layers, as well as their associated uncertainties: (*get\_extparams()*, *get\_intparams()*).

### nt\_typing.py

***NeuralProcessing class***

The *NeuralProcessing* class generates a certain number of values between [average -uncertainty; average + uncertainty] for each of the intensive parameters, scrambles these values, runs the neural networks and collects their results. These steps are made on a per layer basis; this means the neural networks run once for each aerosol layer identified by the *LidarMeasurement* method *get\_layers()*.

Natali.py uses the *NeuralProcessing* class in the following way:

nn=NeuralProcessing(folder=ANN\_FOLDER)

where *ANN\_FOLDER* is the path to the ANNs’ folder; this is the folder that contains all 9 ANN subfolders. The *ANN\_FOLDER* parameter can be set using the command line arguments.

Triggering the processing of a measurement is done with the *process\_measurement()* method:

nn.process\_measurement(measurement,MIN\_ACCEPTED\_CONFIDENCE, FINESSE)

***Election class***

The ***Election*** class is used for the voting system described in the first part of this document. It uses the votes obtained from the ANNs via the *NeuralProcessing* object:

results = Election(votes=votes, min\_ratio= MIN\_AGREEMENT\_PERCENTAGE).results()

The *MIN\_AGREEMENT\_PERCENTAGE* tells the Election object the minimum proportion of answers that need to have a high confidence (higher than *MIN\_ACCEPTED\_CONFIDENCE*) in order to consider the network stable and make its vote eligible.

The value obtained from the *results()* method of the Election object are the determined aerosol types, ordered by layer.

### nt\_output.py

The ***ResultPrinter*** class is used to output both CSV files, as well as human-readable text files. As soon as the results for a given measurement are available, they are stored by the printer object using the *update\_layer()* method. Every time a new measurement is ready to be processed, the ResultPrinter object will flush the data to the disk.

printer=ResultPrinter(True, OUT\_FILENAME)

The *OUT\_FILENAME* parameter tells the software how to name the output files If left at the default value (“auto”), it will name the files after the measurements folder (e.g.: “Bucharest” or “TestData”). For any other value of the parameter, the software will name the output files after the parameter value (useful if you want to integrate the software inside a script).

### nt\_ui.py

***MainWindow class***

The *MainWIndow* class is used to build the entire graphical interface of the program. It contains the toolbar buttons, the text console and the plot figures. They gain functionality by binding certain events to Python functions (for example, an event is generated when the user clicks a button).

Events are also generated from the typing module whenevera new dataset is processed, or when it encounters a warning or error. This is necessary to update the graphical interface with current progress, results of the processing and graph plots.

The MainWindow object represents the core of the entire application and, as such, it is constructed in the *natali.py* script:

main\_window = MainWindow(title=APPLICATION\_TITLE)

***DatasetsDialog class***

The *DatasetsDialog* class is used to display a list of datasets the user can select or deselect for processing. It is constructed whenever the corresponding toolbar button is pressed:

dialog=DatasetsDialog(

 self, title="Choose dataset", datasets=self.\_datasets

)

***SettingsDialog class***

The *SettingsDialog* class is used to display a settings window the user can change the values of certain parameters used by the data and the typing modules. These parameters will be discussed in the software usage chapter.

The settings window is displayed whenever the user presses the corresponding toolbar button:

dialog = SettingsDialog(

 self, title="Settings", settings=self.\_settings

)

## Software requirements

Since the ANNs described above were compiled for 64-bit Windows machines, having a 64-bit Windows Operating System is the main requirement for running the software tool. Any recent version of the operating system (from XP onwards) should suffice.

The software tool is written in Python 2.7 and uses two Python libraries: numpy and netcdf4-python. The latter requires the NetCDF4 C library to be installed in the system. The following list contains all the software requirements:

* Windows XP or newer (must be 64-bit version)
* Python 2.7
* NetCDF4 C library
* numpy Python module
* netcdf4-python module
* Matplotlib Python module
* wxPython Python module

**Note:** This software will only run on 64-bit Windows machines. While you can run the python scripts on 32-bit machines, the ANNs will not run and the software will not produce any results!

## Installation

The following modules for Python should be downloaded and installed:

* Python 2.7
* NetCDF4 C library
* numpy Python module
* netcdf4-python module
* Matplotlib Python module
* wxPython Python module

The five NATALI script files should be copied to the desired location (e.g.: "*C:\NATALI\Software*"). Additionally, the folder containing the 9 ANNs should be copied in the same folder as the Python scripts. However, it is not obligatory to do so, as you can specify the path to the ANNs folder at run time.

The recommend structure of the software's folder is as follows:

* natali.py
* nt\_data.py
* nt\_globals.py
* nt\_output.py
* nt\_typing.py
* nt\_ui.py
* *icons*
	+ error.png
	+ folder.png
	+ page.png
	+ settings.png
	+ start.png
	+ stop.png
	+ success.png
	+ warning.png
* *ANNs*
	+ A1L
	+ A1H
	+ A2L
	+ A2H
	+ A3L
	+ A3H
	+ B1L
	+ B2L
	+ B3L

**Note:** The code contains built-in ANNs, but the user may decide to use its own. However, should the user want to use its own ANNs, the typing module would need to be replaced in order to provide a working interface to the user's ANNs. In this case you should preserve the file names and make a new folder to store the new ANNs. Don’t overwrite the ANNs!

### Python 2.7

Python 2.7 can be downloaded from the official Python website (<https://www.python.org/downloads/>). The latest version as of this time is 2.7.11 and it is the recommended one. To install Python 2.7 simply download and run the installer.

**Note:** In the “*Customize Python 2.7.11*” screen of the installer, make sure to enable “*Add python.exe to Path*” for further ease of use, as shown in the figure below.



Figure 1 Screenshot on the customization of Python installation

If you skip this step, instead of typing “*python*” later you will have to enter the full path of the Python executable (e.g.: *C:\Python27\python.exe*)

### NetCDF4 C library

Download the latest version of the NetCDF C library from the official website (ftp://ftp.unidata.ucar.edu/pub/netcdf ) and run the installer.

**Warning!** Download the appropriate version for your operating system (64-bit). For example, at the time of this writing you would download the “*netCDF4.4.0-NC4-64.exe*” file.

### numpy Python module

The software was built and tested against the 1.10.2 version of the numpy Python module which you can download here: <https://sourceforge.net/projects/numpy/files/NumPy/1.10.2/numpy-1.10.2-win32-superpack-python2.7.exe/download>.

Simply download and run the installer.

### netcdf4-python module

You can get a binary version of the netcdf4-python module as Python .whl file from the following website: <http://www.lfd.uci.edu/~gohlke/pythonlibs/#netcdf4>. To install the .whl Python file, you must run a simple command from the folder where the file is located.

For example, if the file was located in the “*Downloads*” folder of user ”*User*”, you’d have to issue the following commands in a Command Prompt window:

cd C:\Users\User\Downloads

python -m pip install netCDF4-1.2.3.1-cp27-cp27m-win32.whl

**Warning!** Download the 32-bit Python 2.7 version of the module, even if you’re running a 64-bit operating system. As of the time of this writing, this version is called “*netCDF4-1.2.3.1-cp27-cp27m-win32.whl*”.

### Matplotlib module

The Matplotlib module is used to plot the graphs in the Natali application. It can easily be installed using a command similar to the above one:

python -m pip install matplotlib

This will automatically download and install the Matplotlib module and all its dependencies.

**Warning!** To install Matplotlib this way the computer must have a working Internet connection. If this is not possible, you can download a .whl package and install it offline (similar to the netcdf4-python module procedure).

### WxPython module

The wxPython module is used for the graphical interface of the Natali application. To install the module, download the installer (<https://www.wxpython.org/download.php>, “Windows Binaries” section) run it and follow the steps shown on the screen. Download the appropriate version (32- or 64-bit) for Python 2.7!

**Warning!** If you get any errors during the installation, try opening Command Prompt as an administrator and typing the same commands.

## Usage

In order to use the software, you simply need to call the *natali.py* script. The command looks like the following:

python natali.py

It may also be possible to run Natali by double-clicking the *natali.py* file. This is only possible, however, if files with the *.PY* extension are opened by default with python. Once you run Natali you will see the main window of the application:

Figure 1 Screenshot of the software’s main window

The graphical interface is structured in four parts:

* **the toolbar** – located in top part of the window, it contains buttons to control the software
* **the console** – located in the top left part of the window, the software will output messages here
* **the timeseries plot** – located in the top right part of the window, here you can see an overview of all the processed datasets, allowing you to observe the evolution of aerosol layers in time
* **the data plots** – located in the bottom half of the window, these display the extensive parameters, the computed intensive parameters and the determined aerosol layers.

To start using the software, the user must first choose a folder containing the data. Pressing the “Choose data folder” will open a “Choose data folder” dialog, where the user can select the said folder.

Figure 1 Screenshot of the “Choose data folder” dialog

This will enable the “Choose datasets” toolbar button, which allows the user to select certain datasets inside the data folder.

Figure 1 Screenshot of the “Choose datasets” dialog

The number of selected datasets is displayed in two places: in the toolbar button and in the status bar, right at the bottom of the window. Once the processing is started, all the selected datasets will be processed; this means the list of datasets cannot be changed once the processing is started.

The Natali software provides a method to stop the processing by means of a “Stop” button inside the toolbar, whenever the processing is taking place. Note, however, that the software will not stop when the ANNs are running so you may experience some delay (up to 1-2 minutes) between clicking the “Stop” button and the processing actually stopping.

Figure 1 Screenshot of the software while running

After all the selected datasets have been processed, the user will be presented with a timeseries plot of the detected aerosol layers inde all the processed datasets; this provides a method of displaying the evolution of aerosol layers for consecutive measurements.

If the user wants to visualize data for a certain dataset, a dropdown menu situated below the console allows precisely that (right next to “View data for:”). The extensive parameters, the computed parameters and the determined aerosol layers will be displayed in the four bottom plots.

It is also possible to save any plot as a .PNG image. To do this, the user has to right-click on the desired plot and select the “Save as...” option. A prompt will be displayed on the screen, asking for the location and name of the file you want to save. The saved file will be a high-resolution of the plot (the image will be identical to what is displayed in the software, except it will have a much-higher resolution). The increased resolution allows the image to be used in documents, web and event print materials.



Figure 1 Screenshot on saving the plots as image files

Before starting the processing, the user can also adjust several parameters which can alter the way the software runs. These are presented in the Settings window, which can be opened with the help of the “Settings” toolbar button.

Figure 1 Screenshot on the running parameter customization

The full list of user-changeable parameters is listed in the table below, together with their effect on the software and their default values:

|  |  |  |
| --- | --- | --- |
| **Argument name** | **Role** | **Default value** |
| Altitude scale | These are the minimum and maximum value on the vertical (altitude) axis. You can change this either before or after processing. | 0-5 |
| ANN Folder | This is the folder containing all the ANNs' folders (A1L, A1H, A2L, etc.) | "ANNs" |
| Filter window | Enhances the smoothing effect of the derivative; has an impact on the finesse of layer structures identified. Use values between 500 - 1000 | 700 |
| Minimum layer depth (m) | Rejects layer sub-structures thinner than the selected value (in metres). Use values between 200 and 500. | 300 |
| Finesse | Number of values generated between [value – uncertainty] and [value + uncertainty] for each intensive optical parameter; adds statistical significance to the cases presented to the ANNs Use values between 10 and 50. | 20 |
| Minimum accepted confidence (%) | Threshold of confidence above which the answers from an ANN is accepted Use values between 0.6 and 0.8. | 70 |
| Minimum agreement ratio (%) | Threshold of number of agreements above which the answer from an ANN is considered relevant and passed to the voting Use values between 0.2 and 0.5. | 25 |

Table 1 List of the accepted command line arguments, examples and explanations

## Output files

Following the execution of the script, three output files will be written:

* A telegram (human-readable text file)
* A .CSV file
* A log file

The names under which the first three will be created is described above, in the *nt\_output.py* section. The name of the log file is "*natali\_log.txt*".

### The log file

The log file is meant to provide a quick overview of the processing runs. For each run of the software, it specifies the folder used to search for measurement files, the measurements that were processed and the output file paths. The log file looks like below:

--------------------------------

Start run time: 2016-03-27 15:54

--------------------------------

Input

-----

Data folder: C:\Users\User\Desktop\Work\Natali\Data\Measured\Barcelona

Measurements:

 ba1207100300

Output

------

CSV file path: C:\Users\User\Desktop\Work\Natali\Barcelona.csv

Report file path: C:\Users\User\Desktop\Work\Natali\Barcelona.txt

### The .TXT file

The text file contains telegrams from the software (e.g.: errors, progress reports), as well as the computation results. A list of possible messages is provided below.

|  |  |
| --- | --- |
| **Comment** | **Reason** |
| Typing not possible: intensive parameter [...] cannot be calculated | One or more of the extensive parameters are not available at the layer altitude |
| Typing not possible: values of the intensive parameter [...] out of acceptable range | One or more intensive parameters have values which are physically impossible (see Table 1) |
| Typing not possible: no ANN passed the confidence criteria | None of the voting ANNs reached the 70% confidence for none of the aerosol types; the optical data are probably not well calibrated |
| Typing not possible: no ANN passed the minimum agreement criteria | None of the voting ANNs reached the 25% agreement for the cases generated between the error bars; the optical data are probably not well calibrated |
| Typing uncertain: relative error of intensive parameters [...] higher than 20% | One or more intensive parameters have large uncertainties, higher than accepted by the ANNs; the answer may not be correct, as the ANNs are “guessing” |

Table 1 List of messages possible in the telegrams

Each measurement is clearly delimited by a header containing the measurement name. Each measurement should contain one or more layers. An example output is shown below:

--------------------------------

Start run time: 2016-03-27 15:54

--------------------------------

Input

-----

Data folder: C:\Users\User\Desktop\Work\Natali\Data\Measured\Barcelona

Measurements:

 ba1207100300

Run parameters

------------------

FILTER\_WINDOW: 700

MIN\_LAYER\_DEPTH: 300

AVERAGING\_LAYER\_DEPTH: 200

FINESSE: 10

ANN\_FOLDER: C:\Users\User\Desktop\Work\Natali\ANNs

MIN\_ACCEPTED\_CONFIDENCE: 0.700000

MIN\_AGREEMENT\_PERCENTAGE: 0.250000

+------------------------------------+

| ba1207100300 |

+------------------------------------+

Layer 1:

 Bottom: 424.0 [m]

 Top: 798.0 [m]

 Retrieval Bottom: 424.0 [m]

 Retrieval Top: 798.0 [m]

 AE355\_532: 3.69

 AE355\_532\_ERR: 0.00

 CI355\_532: 1.86

 CI355\_532\_ERR: 0.00

 CI532\_1064: 1.81

 CI532\_1064\_ERR: 0.06

 CR355\_532: 2.89

 CR355\_532\_ERR: 0.00

 CR532\_1064: 4.27

 CR532\_1064\_ERR: 0.25

 LR355: 16 [sr]

 LR355\_ERR: 0.00 [sr]

 LR532: 11 [sr]

 LR532\_ERR: 0.00 [sr]

 DEP532: N/A

 DEP532\_ERR: N/A

 Predominant\_Component: Continental

 Aerosol\_Type: Unknown

 Comments:

 A1L\_Answer: N/A

 A1L\_Confidence: 0

 A1L\_Agreements: 0

 A1H\_Answer: N/A

 A1H\_Confidence: 0

 A1H\_Agreements: 0

 A2L\_Answer: N/A

 A2L\_Confidence: 0

 A2L\_Agreements: 0

 A2H\_Answer: N/A

 A2H\_Confidence: 0

 A2H\_Agreements: 0

 A3L\_Answer: N/A

 A3L\_Confidence: 0

 A3L\_Agreements: 0

 A3H\_Answer: N/A

 A3H\_Confidence: 0

 A3H\_Agreements: 0

 B1L\_Answer: Continental

 B1L\_Confidence: 0.99

 B1L\_Agreements: 5000

 B2L\_Answer: Dust

 B2L\_Confidence: 0.76

 B2L\_Agreements: 5000

 B3L\_Answer: N/A

 B3L\_Confidence: 0

 B3L\_Agreements: 0

### The .CSV file

The .CSV file is a simple comma-separated values file with a header line. Each subsequent line represents an aerosol layer. The data written on each line of the .CSV file is the same as the data written to the .TXT file for each layer.

The order of the columns written to the .CSV file are described in the following table:

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Index** | **Column Name** | **Description** | **Example value** |
| 0 | Bottom | Altitude of the bottom of the layer in meters. | 803 |
| 1 | Top | Altitude of the top of the layer in meters. | 1221 |
| 2 | Retrieval Bottom | Lowest altitude of the retrieval area for this layer. This is the area inside which the typing is performed, as it has a high signal-to-noise ratio. | 803.2 |
| 3 | Retrieval Top | Highest altitude of the retrieval area for this layer. This is the area inside which the typing is performed, as it has a high signal-to-noise ratio. | 1146.75 |
| 4 | AE355\_550 | Angstrom Exponent | -0.35 |
| 5 | AE355\_532\_ERR | Angstrom Exponent Absolute Error | 0.10 |
| 6 | CI355\_532 | Color Index (355nm/532nm) | -0.61 |
| 7 | CI355\_532\_ERR | Color Index (355nm/532nm) Absolute Error | 0.10 |
| 8 | CI532\_1064 | Color Index (532nm/1064nm) | -0.51 |
| 9 | CI532\_1064\_ERR | Color Index (532nm/1064nm) Absolute Error | 0.06 |
| 10 | CR355\_532 | Color Ratio (355nm/532nm) | 0.78 |
| 11 | CR350\_532\_ERR | Color Ratio (355nm/532nm) Absolute Error | 0.03 |
| 12 | CR550\_1000 | Color Ratio (532nm/1064nm) | 0.70 |
| 13 | CR550\_1000\_ERR | Color Ratio (532nm/1064nm) Absolute Error | 0.03 |
| 14 | LR355 | Lidar Ratio (355nm) in sr | 44 |
| 15 | LR355\_ERR | Lidar Ratio (355nm) Absolute Error in sr | 2 |
| 16 | LR532 | Lidar Ratio (532nm) in sr | 40 |
| 17 | LR532\_ERR | Lidar Ratio (532nm) Absolute Error in sr | 2 |
| 18 | DEP532 | Linear Particle Depolarization Ratio (532nm) | 0.28 |
| 19 | DEP532\_ERR | Linear Particle Depolarization Ratio (532nm) Absolute Error | 0.04 |
| 20 | Predominant\_Aerosol | Low resolution aerosol type  | Dust |
| 21 | Aerosol\_Type | High resolution aerosol type | Dust |
| 22 | Comments | Remarks about the aerosol layer | Typing uncertain; relative error of intensive parameters ['lr\_532', 'lr\_355'] higher than 20% |
| 23 | A1L\_Answer | A1L ANN aerosol type | Dust |
| 24 | A1L\_Confidence | A1L ANN confidence level | 0.88 |
| 25 | A1L\_Agreements | A1L ANN confident answers | 6000 |
| 26 | A1H\_Answer | A1H ANN aerosol type | Dust |
| 27 | A1H\_Confidence | A1H ANN confidence level | 0.96 |
| 28 | A1H\_Agreements | A1H ANN confident answers | 6000 |
| 29 | A2L\_Answer | A2L ANN aerosol type | Dust |
| 30 | A2L\_Confidence | A2L ANN confidence level | 0.94 |
| 31 | A2L\_Agreements | A2L ANN confident answers | 6000 |
| 32 | A2H\_Answer | A2H ANN aerosol type | Dust |
| 33 | A2H\_Confidence | A2H ANN confidence level | 0.78 |
| 34 | A2H\_Agreements | A2H ANN confident answers | 5913 |
| 35 | A3L\_Answer | A3L ANN aerosol type | Dust |
| 36 | A3L\_Confidence | A3L ANN confidence level | 0.88 |
| 37 | A3L\_Agreements | A2L ANN confident answers | 5928 |
| 38 | A3H\_Answer | A3H ANN aerosol type | Dust |
| 39 | A3H\_Confidence | A3H ANN confidence level | 0.91 |
| 40 | A3H\_Agreements | A3H ANN confident answers | 5994 |
| 41 | B1L\_Answer | B1L ANN aerosol type | Marine |
| 42 | B1L\_Confidence | B1L ANN confidence level | 0.83 |
| 43 | B1L\_Agreements | B1L ANN confident answers | 5945 |
| 44 | B2L\_Answer | B2L ANN aerosol type | N/A |
| 45 | B2L\_Confidence | B2L ANN confidence level | 0 |
| 46 | B2L\_Agreements | B2L ANN confident answers | 0 |
| 47 | B3L\_Answer | B3L ANN aerosol type | ContinentalPolluted |
| 48 | B3L\_Confidence | B3L ANN confidence level | 0.73 |
| 49 | B3L\_Agreements | B3L ANN confident answers | 5877 |

Table 1 List of parameters in the .csv file

# Datasheet on software performances, limitations and constraints

| **Parameter** | **Performances** | **Limitations** | **Constraints** |
| --- | --- | --- | --- |
| Operating environment | - | requires Windows operating system to run the ANNs;requires write permissions to its own folder (may not work inside C:\Program Files or C:\Windws) | requires:* Python 2.7 (does not work with Python 3)
* NetCDF4 library version 4.4.0-NC4
* NumPy Python module version 1.10.2
* netcdf4-python module version 1.2.3.1 for 32 bits
 |
| Input datasets | EARLINET NetCDF files | detection wavelengths: 1064, 532, 355 nm | simultaneous provision of:* backscatter coefficient 1064 nm
* backscatter coefficient 532 nm
* backscatter coefficient 355 nm
* extinction coefficient 532 nm
* extinction coefficient 355 nm
 |
| Layer detection | Layers thicker than 300 m | PBL cannot be retrieved if the overlap of the lidar is below the top of the PBL | differences in smoothing of the 3 and 2 affects the value of the intensive parameters >> only regions within the layer with a SNR > 5 are considered to calculate mean layer values |
| Angstrom coefficient | higher typing confidence for relative error ≤ 20% | no typing possible if this parameter is missing or out of range (-2 … 6) | no constraint, flag if relative error higher than 20% |
| Color ratio | higher typing confidence for relative error ≤ 20% | no typing possible if this parameter is missing or out of range (-2 … 6) | no constraint, flag if relative error higher than 20% |
| Color index | higher typing confidence for relative error ≤ 20% | no typing possible if this parameter is missing or out of range (-2 … 6) | no constraint, flag if relative error higher than 20% |
| Lidar ratio (sr) | higher typing confidence for relative error ≤ 20% | no typing possible if this parameter is missing or out of range (5 …200 sr) | no constraint, flag if relative error higher than 20% |
| Linear particle depolarization ratio (%) | higher typing confidence for relative error ≤ 20% | only low resolution typing (predominant aerosol type) possible if this parameter is missing or out of range (0 … 0.6) | no constraint, flag if relative error higher than 20% |
| Percentage of recognition of the aerosol type by vote of A1H, A2H, A3H  | Continental: 100.0% | optical data well calibratedrelative error of all intensive optical parameters ≤ 20%;LPDR available | *FILTER\_WINDOW* = 700*MIN\_LAYER\_DEPTH* = 300*FINESSE* = 20*MIN\_ACCEPTED\_CONFIDENCE* = 0.7*MIN\_AGREEMENT\_RATIO* = 0.25 |
| Dust: 100.0% |
| Continental polluted: 85.7% |
| Marine: 100.0% |
| Smoke: 100.0% |
| Volcanic: 100.0% |
| Continental dust: 100.0% |
| Marine mineral: 100.0% |
| Continental smoke: 76.2% |
| Dust polluted: 95.2% |
| Coastal: 85.7% |
| Coastal polluted: 76.2% |
| Mixed dust: 90.5% |
| Mixed smoke: 100.0% |
| Percentage of recognition of the predominant aerosol type by vote of A1L, A2L, A3L  | Continental: 100.0% | optical data well calibratedrelative error of all intensive optical parameters ≤ 20%;LPDR available | *FILTER\_WINDOW* = 700*MIN\_LAYER\_DEPTH* = 300*FINESSE* = 20*MIN\_ACCEPTED\_CONFIDENCE* = 0.7*MIN\_AGREEMENT\_RATIO* = 0.25Mixtures are considered recognized if at least one of their components is recognized  |
| Dust: 100.0% |
| Continental polluted: 100.0% |
| Marine: 100.0% |
| Smoke: 100.0% |
| Volcanic: 100.0% |
| Continental dust (continental / dust): 95.2% |
| Marine mineral (marine / dust / volcanic): 100.0% |
| Continental smoke (continental / smoke): 76.2% |
| Dust polluted (dust / smoke): 95.2% |
| Coastal (marine / continental): 100.0% |
| Coastal polluted (marine / continental polluted): 71.4% |
| Mixed dust (dust / continental / marine): 100.0% |
| Mixed smoke (smoke / continental / marine): 81.8% |
| Percentage of recognition of the predominant aerosol type by vote of B1L, B2L, B3L | Continental: 100.0% | optical data well calibratedrelative error of all intensive optical parameters ≤ 20%;Volcanic cannot be retrieved because LPDR is not available (overlap of the spectral parameters with Dust and/or Continental polluted) | *FILTER\_WINDOW* = 700*MIN\_LAYER\_DEPTH* = 300*FINESSE* = 20*MIN\_ACCEPTED\_CONFIDENCE* = 0.7*MIN\_AGREEMENT\_RATIO* = 0.25Mixtures are considered recognized if at least one of their components is recognized  |
| Dust: 100.0% |
| Continental polluted: 90.5% |
| Marine: 100.0% |
| Smoke: 100.0% |
| Continental dust (continental / dust): 95.2% |
| Marine mineral (marine / dust): 100.0% |
| Continental smoke (continental / smoke): 52.4% |
| Dust polluted (dust / smoke): 57.1% |
| Coastal (marine / continental): 95.2% |
| Coastal polluted (marine / continental polluted): 52.4% |
| Mixed dust (dust / continental / marine): 100.0% |
| Mixed smoke (smoke / continental / marine): 68.2% |
| run time per layer (with LPDR) | ~7s |  | *Tested on:** Intel Core i5-4460
* 16GB of RAM
* 500GB SSD
* Windows 10
 |
| run time per layer (without LPDR) | ~4s |  | *Tested on:** Intel Core i5-4460
* 16GB of RAM
* 500GB SSD
* Windows 10
 |

# Copyright and disclaimer notice

## Copyright notice

The NATALI software is developed in the framework of the NATALI 4000110671/14/I-LG ESA/ESRIN contract, and is the property of the National Institute of R&D for Optoelectronics (INOE).

## Disclaimer notice

The information provided in this document is intended for informational purposes only and is subject to change, following improvements and/or changes in the NATALI software.

The NATALI software is provided ``as is'' and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. INOE is not responsible for the quality of the optical data used for classification, which may lead to incorrect typing, especially if the calibration is not appropriate.

In no event shall INOE be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

## Software usage policy

Redistribution and use in source and binary forms of the NATALI software, with or without modification, are permitted provided that the following conditions are met:

* Redistributions must reproduce the above copyright notice, this list of conditions and the disclaimer notice in the documentation and/or other materials provided with the distribution. Redistributions of source code must also reproduce this information in the source code itself.
* If the program is modified, redistributions must include a notice (in the same places as above) indicating that the redistributed program is not identical to the version distributed by INOE.
* All advertising materials mentioning features or use of this software must display the following acknowledgment: “This product includes software developed by the National Institute of R&D for Optoelectronics (INOE).”
* In case the software is used to produce publishable results, the authors are kindly asked to cite: Nicolae D., Vasilescu J., Talianu C., Dandocsi A., Independent retrieval of aerosol type from lidar, *EPJ Web of Conferences (*[*www.epj-conferences.org*](http://www.epj-conferences.org/)*)*, in press
* The name of INOE may not be used to endorse or promote products derived from this software without specific prior written permission.